

PCT/JP 99/01394

ESKU 日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

19.03.99

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application:

1998年 3月20日

REC'D 17 MAY 1999

出願番号
Application Number:

平成10年特許願第071838号

WIPO PCT

出願人
Applicant(s):

株式会社アイエスデー研究所

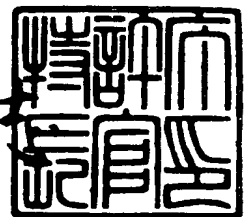
PRIORITY
DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

1999年 4月23日

特許庁長官
Commissioner,
Patent Office

伴佐山 建志



出証番号 出証特平11-3024558

【書類名】 特許願

【整理番号】 X-9032

【提出日】 平成10年 3月20日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 1/00

【発明の名称】 ソフトウェアの生成装置及びソフトウェアの生成方法

【請求項の数】 2

【発明者】

 【住所又は居所】 神奈川県鎌倉市十二所 9 6 7 - 6 4

 【氏名】 根来 文生

【発明者】

 【住所又は居所】 千葉県山武郡成東町寺崎 5 4 8

 【氏名】 戸村 茂昭

【特許出願人】

 【住所又は居所】 神奈川県鎌倉市十二所 9 6 7 - 6 4

 【氏名又は名称】 根来 文生

【代理人】

 【識別番号】 100077849

 【弁理士】

 【氏名又は名称】 須山 佐一

 【電話番号】 03-3254-1039

【手数料の表示】

 【納付方法】 予納

 【予納台帳番号】 014395

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

特平 10-071838

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ソフトウェアの生成装置及びソフトウェアの生成方法

【特許請求の範囲】

【請求項1】 (a) 生産するソフトウェアに必要な画面、ファイルまたは帳票（以下、画面、ファイル及び帳票を「定義体」と呼ぶ。）に存在する全ての単語に対して作成され、かつ定義体、単語それぞれに対する定義体識別子及び単語識別子が付された、画面編集を行うためのW04位相要素及びW04論理要素、処理経路を決定するためのW02論理要素、並びにファイル更新のデータ編集を行うためのW03位相要素及びW03論理要素であって、

(a-1) W04位相要素は、同じ単語のW03のデータエリアが”空”でない場合には、当該W04の当該データエリアに前記同じ単語のW03のデータエリアの値を設定するものであり、

(a-2) W04論理要素は、当該単語のW04のデータエリアが”空”でない場合であって同じ単語のW02またはW03の不成立フラグがセットされている場合には、不成立コードをセットするとともに不成立フラグをリセットし、当該単語のW04のデータエリアが“空の場合”には当該単語以外の単語のW04のデータエリアから当該単語の設定値を導出して当該設定値を当該単語のW04のデータエリアに設定し、前記設定値を導出できない場合であって再起動可能なときには再起動フラグをセットし、再起動不可能なときには不成立コードをセットするものであり、

(a-3) W02論理要素は、当該単語のW02のデータエリアが”空”でない場合には、処理経路フラグを設定し、処理経路フラグ設定が不成立な場合であって再起動可能なときには再起動フラグをセットし、再起動不可能なときには不成立フラグをセットするものであり、

(a-4) W03位相要素は、同じ単語のW02のデータエリアが”空”でない場合には、当該W03の当該データエリアに前記同じ単語のW02のデータエリアの値を設定するものであり、

(a-5) W03論理要素は、当該単語のW03のデータエリアが”空”の場合には、当該単語以外の単語のW03のデータエリアから当該単語の設定値を導出

して当該設定値を当該単語のW03のデータエリアに設定し、前記設定値を導出できない場合であって再起動可能なときには再起動フラグをセットし、再起動不可能なときには不成立フラグをセットするもの、

を有する手段と、

(b) 画面毎に当該画面上に存在する単語の各W04位相要素及びW04論理要素を位相要素群、論理要素群の順番に実行可能にまとめるとともに各単語の各W04データエリアを組み込んだW04パレット、画面毎に当該画面上に存在する単語の各W02論理要素を実行可能にまとめるとともに各単語の各W02データエリアを組み込んだW02パレット、及びシステム内の全定義体上に存在する単語の各W03位相要素及びW03論理要素を位相要素群、論理要素群の順番に実行可能にまとめるとともに各単語の各W03データエリアを組み込んだW03パレットを有するパレット群と、

(c) 各パレット内の各位相要素及び論理要素を実行すると共に、当該パレット内のパレット再起動フラグがセットされている場合には、当該パレット内の各位相要素及び論理要素を再実行させる手段と、

(d) 前記W04パレットの実行に基づく画面データをコンピュータの画面上に送信し、該送信画面に対する応答画面データを受信して前記W02パレットを実行し、当該W02パレットの前記処理経路フラグに基づき処理経路を決定し、決定された処理経路に応じた前記W03パレットを実行する手段と

を具備するソフトウェアを生成する装置であって、

前記定義体識別子、前記単語識別子、前記処理経路及び前記設定値を導出するための導出データが未定義のW04位相要素、W04論理要素、W02論理要素、W03位相要素、W03論理要素及び上記(b)のパレット群並びに(c)の手段を予め記憶する手段と、

前記単語毎に前記定義体識別子、前記単語識別子、前記処理経路及び前記導出データを入力する手段と、

前記単語毎に前記入力された前記定義体識別子、前記単語識別子、前記処理経路及び前記導出データを、前記記憶された未定義のW04位相要素、W04論理要素、W02論理要素、W03位相要素、W03論理要素及び上記の(b)のパ

レット群並びに(c)の手段の所定の位置に埋め込む手段と
を具備することを特徴とするソフトウェアの生成装置。

【請求項2】(a)生産するソフトウェアに必要な画面、ファイルまたは帳票
(以下、画面、ファイル及び帳票を「定義体」と呼ぶ。)に存在する全ての単語
に対して作成され、かつ定義体、単語それぞれに対する定義体識別子及び単語識
別子が付された、画面編集を行うためのW04位相要素及びW04論理要素、処
理経路を決定するためのW02論理要素、並びにファイル更新のデータ編集を行
うためのW03位相要素及びW03論理要素であって、

(a-1)W04位相要素は、同じ単語のW03のデータエリアが”空”でない
場合には、当該W04の当該データエリアに前記同じ単語のW03のデータエリ
アの値を設定するものであり、

(a-2)W04論理要素は、当該単語のW04のデータエリアが”空”でない
場合であって同じ単語のW02またはW03の不成立フラグがセットされている
場合には、不成立コードをセットするとともに不成立フラグをリセットし、当該
単語のW04のデータエリアが“空の場合”には当該単語以外の単語のW04の
データエリアから当該単語の設定値を導出して当該設定値を当該単語のW04の
データエリアに設定し、前記設定値を導出できない場合であって再起動可能なと
きには再起動フラグをセットし、再起動不可能なときには不成立コードをセット
するものであり、

(a-3)W02論理要素は、当該単語のW02のデータエリアが”空”でない
場合には、処理経路フラグを設定し、処理経路フラグ設定が不成立な場合であっ
て再起動可能なときには再起動フラグをセットし、再起動不可能なときには不成立
フラグをセットするものであり、

(a-4)W03位相要素は、同じ単語のW02のデータエリアが”空”でない
場合には、当該W03の当該データエリアに前記同じ単語のW02のデータエリ
アの値を設定するものであり、

(a-5)W03論理要素は、当該単語のW03のデータエリアが”空”の場合
には、当該単語以外の単語のW03のデータエリアから当該単語の設定値を導出
して当該設定値を当該単語のW03のデータエリアに設定し、前記設定値を導出

できない場合であって再起動可能なときには再起動フラグをセットし、再起動不可能なときには不成立フラグをセットするもの、
を有する手段と、

(b) 画面毎に当該画面上に存在する単語の各W04位相要素及びW04論理要素を位相要素群、論理要素群の順番に実行可能にまとめるとともに各単語の各W04データエリアを組み込んだW04パレット、画面毎に当該画面上に存在する単語の各W02論理要素を実行可能にまとめるとともに各単語の各W02データエリアを組み込んだW02パレット、及びシステム内の全定義体上に存在する単語の各W03位相要素及びW03論理要素を位相要素群、論理要素群の順番に実行可能にまとめるとともに各単語の各W03データエリアを組み込んだW03パレットを有するパレット群と、

(c) 各パレット内の各位相要素及び論理要素を実行すると共に、当該パレット内のパレット再起動フラグがセットされている場合には、当該パレット内の各位相要素及び論理要素を再実行させる手段と、

(d) 前記W04パレットの実行に基づく画面データをコンピュータの画面上に送信し、該送信画面に対する応答画面データを受信して前記W02パレットを実行し、当該W02パレットの前記処理経路フラグに基づき処理経路を決定し、決定された処理経路に応じた前記W03パレットを実行する手段と

を具備するソフトウェアを生成する方法であって、
前記定義体識別子、前記単語識別子、前記処理経路及び前記設定値を導出するための導出データが未定義のW04位相要素、W04論理要素、W02論理要素、W03位相要素、W03論理要素及び上記の(b)のパレット群並びに(c)の手段を予め記憶しておき、

前記単語毎に前記定義体識別子、前記単語識別子、前記処理経路及び前記導出データを入力し、

前記単語毎に前記入力された前記定義体識別子、前記単語識別子、前記処理経路及び前記導出データを、前記記憶された未定義のW04位相要素、W04論理要素、W02論理要素、W03位相要素、W03論理要素及び上記の(b)のパレット群並びに(c)の手段の所定の位置に埋め込む

ことを特徴とするソフトウェアの生成方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、業務用ソフトウェアやゲーム用ソフトウェア、その他あらゆる分野のソフトウェアに適用されるソフトウェアの生成装置及びソフトウェアの生成方法に関する。

【0002】

【従来の技術】

従来のソフトウェアは、要件定義、基本設計、詳細設計、プログラミング、コンパイル、プログラムテスト及び総合テストを経て作られている。これらの一連の工程は、SEの経験、知識、応用力等の属人的能力によって行われる。

【0003】

従って、現実的にはソフトウェアを自動的に生成しようとする発想自体が無謀であり、その目的は当該分野において非予測的であるといえる。

【0004】

【発明が解決しようとする課題】

にもかかわらず、本発明は、ソフトウェアをSEの経験、知識、応用力等の属人的能力を必要とせず機械的アルゴリズムだけで自動的に生成することができるソフトウェアの生成装置及びソフトウェアの生成方法を提供することを目的とする。

【0005】

これを可能とする本発明の背景は、普遍的なソフトウェア、即ち要件事象に対して唯一的に存在するソフトウェアの決定が可能なLYEEの定理を前提としてはじめて生成することができるものであり、その結果、生産性の向上、品質の均一化及び保守の容易化を図ることができるソフトウェアの生成装置及びソフトウェアの生成方法を提供することが可能となる。

【0006】

【課題を解決するための手段】

本発明は、上記目的を達成するためになされたものである。

【0007】

本発明では、(a) 生産するソフトウェアに必要な画面、ファイルまたは帳票（以下、画面、ファイル及び帳票を「定義体」と呼ぶ。）に存在する全ての単語に対して作成され、かつ定義体、単語それぞれに対する定義体識別子及び単語識別子が付された、画面編集を行うためのW04位相要素及びW04論理要素、処理経路を決定するためのW02論理要素、並びにファイル更新のデータ編集を行うためのW03位相要素及びW03論理要素であって、

(a-1) W04位相要素は、同じ単語のW03のデータエリアが”空”でない場合には、当該W04の当該データエリアに前記同じ単語のW03のデータエリアの値を設定するものであり、

(a-2) W04論理要素は、当該単語のW04のデータエリアが”空”でない場合であって同じ単語のW02またはW03の不成立フラグがセットされている場合には、不成立コードをセットするとともに不成立フラグをリセットし、当該単語のW04のデータエリアが“空の場合”には当該単語以外の単語のW04のデータエリアから当該単語の設定値を導出して当該設定値を当該単語のW04のデータエリアに設定し、前記設定値を導出できない場合であって再起動可能などときには再起動フラグをセットし、再起動不可能などときには不成立コードをセットするものであり、

(a-3) W02論理要素は、当該単語のW02のデータエリアが”空”でない場合には、処理経路フラグを設定し、処理経路フラグ設定が不成立な場合であって再起動可能などときには再起動フラグをセットし、再起動不可能などときには不成立フラグをセットするものであり、

(a-4) W03位相要素は、同じ単語のW02のデータエリアが”空”でない場合には、当該W03の当該データエリアに前記同じ単語のW02のデータエリアの値を設定するものであり、

(a-5) W03論理要素は、当該単語のW03のデータエリアが”空”の場合

には、当該単語以外の単語のW03のデータエリアから当該単語の設定値を導出して当該設定値を当該単語のW03のデータエリアに設定し、前記設定値を導出できない場合であって再起動可能なときには再起動フラグをセットし、再起動不可能なときには不成立フラグをセットするもの、
を有する手段と、

(b) 画面毎に当該画面上に存在する単語の各W04位相要素及びW04論理要素を位相要素群、論理要素群の順番に実行可能にまとめるとともに各単語の各W04データエリアを組み込んだW04パレット、画面毎に当該画面上に存在する単語の各W02論理要素を実行可能にまとめるとともに各単語の各W02データエリアを組み込んだW02パレット、及びシステム内の全定義体上に存在する単語の各W03位相要素及びW03論理要素を位相要素群、論理要素群の順番に実行可能にまとめるとともに各単語の各W03データエリアを組み込んだW03パレットを有するパレット群と、

(c) 各パレット内の各位相要素及び論理要素を実行すると共に、当該パレット内のパレット再起動フラグがセットされている場合には、当該パレット内の各位相要素及び論理要素を再実行させる手段と、

(d) 前記W04パレットの実行に基づく画面データをコンピュータの画面上に送信し、該送信画面に対する応答画面データを受信して前記W02パレットを実行し、当該W02パレットの前記処理経路フラグに基づき処理経路を決定し、決定された処理経路に応じた前記W03パレットを実行する手段と

を具備するソフトウェアを生成する装置であって、
前記定義体識別子、前記単語識別子、前記処理経路及び前記設定値を導出するための導出データが未定義のW04位相要素、W04論理要素、W02論理要素、W03位相要素、W03論理要素及び上記(b)のパレット群並びに(c)の手段を予め記憶する手段と、

前記単語毎に前記定義体識別子、前記単語識別子、前記処理経路及び前記導出データを入力する手段と、

前記単語毎に前記入力された前記定義体識別子、前記単語識別子、前記処理経路及び前記導出データを、前記記憶された未定義のW04位相要素、W04論理

要素、W02論理要素、W03位相要素、W03論理要素及び上記の(b)のパレット群並びに(c)の手段の所定の位置に埋め込む手段と
を具備することを特徴とするソフトウェアの生成装置、が提供される。

【0008】

また、本発明では、(a)生産するソフトウェアに必要な画面、ファイルまたは帳票(以下、画面、ファイル及び帳票を「定義体」と呼ぶ。)に存在する全ての単語に対して作成され、かつ定義体、単語それぞれに対する定義体識別子及び単語識別子が付された、画面編集を行うためのW04位相要素及びW04論理要素、処理経路を決定するためのW02論理要素、並びにファイル更新のデータ編集を行うためのW03位相要素及びW03論理要素であって、

(a-1) W04位相要素は、同じ単語のW03のデータエリアが”空”でない場合には、当該W04の当該データエリアに前記同じ単語のW03のデータエリアの値を設定するものであり、

(a-2) W04論理要素は、当該単語のW04のデータエリアが”空”でない場合であって同じ単語のW02またはW03の不成立フラグがセットされている場合には、不成立コードをセットするとともに不成立フラグをリセットし、当該単語のW04のデータエリアが“空の場合”には当該単語以外の単語のW04のデータエリアから当該単語の設定値を導出して当該設定値を当該単語のW04のデータエリアに設定し、前記設定値を導出できない場合であって再起動可能なときには再起動フラグをセットし、再起動不可能なときには不成立コードをセットするものであり、

(a-3) W02論理要素は、当該単語のW02のデータエリアが”空”でない場合には、処理経路フラグを設定し、処理経路フラグ設定が不成立な場合であって再起動可能なときには再起動フラグをセットし、再起動不可能なときには不成立フラグをセットするものであり、

(a-4) W03位相要素は、同じ単語のW02のデータエリアが”空”でない場合には、当該W03の当該データエリアに前記同じ単語のW02のデータエリアの値を設定するものであり、

(a-5) W03論理要素は、当該単語のW03のデータエリアが”空”の場合

には、当該単語以外の単語のW03のデータエリアから当該単語の設定値を導出して当該設定値を当該単語のW03のデータエリアに設定し、前記設定値を導出できない場合であって再起動可能なときには再起動フラグをセットし、再起動不可能なときには不成立フラグをセットするもの、
を有する手段と、

(b) 画面毎に当該画面上に存在する単語の各W04位相要素及びW04論理要素を位相要素群、論理要素群の順番に実行可能にまとめるとともに各単語の各W04データエリアを組み込んだW04パレット、画面毎に当該画面上に存在する単語の各W02論理要素を実行可能にまとめるとともに各単語の各W02データエリアを組み込んだW02パレット、及びシステム内の全定義体上に存在する単語の各W03位相要素及びW03論理要素を位相要素群、論理要素群の順番に実行可能にまとめるとともに各単語の各W03データエリアを組み込んだW03パレットを有するパレット群と、

(c) 各パレット内の各位相要素及び論理要素を実行すると共に、当該パレット内のパレット再起動フラグがセットされている場合には、当該パレット内の各位相要素及び論理要素を再実行させる手段と、

(d) 前記W04パレットの実行に基づく画面データをコンピュータの画面上に送信し、該送信画面に対する応答画面データを受信して前記W02パレットを実行し、当該W02パレットの前記処理経路フラグに基づき処理経路を決定し、決定された処理経路に応じた前記W03パレットを実行する手段と

を具備するソフトウェアを生成する方法であって、
前記定義体識別子、前記単語識別子、前記処理経路及び前記設定値を導出するための導出データが未定義のW04位相要素、W04論理要素、W02論理要素、W03位相要素、W03論理要素及び上記の(b)のパレット群並びに(c)の手段を予め記憶しておき、

前記単語毎に前記定義体識別子、前記単語識別子、前記処理経路及び前記導出データを入力し、

前記単語毎に前記入力された前記定義体識別子、前記単語識別子、前記処理経路及び前記導出データを、前記記憶された未定義のW04位相要素、W04論理

要素、W02論理要素、W03位相要素、W03論理要素及び上記の(b)のパレット群並びに(c)の手段の所定の位置に埋め込む

ことを特徴とするソフトウェアの生成方法、が提供される。

【0009】

本発明者等は、新規なソフトウェアの生産方法を既に提唱している(1件は国際公開番号WO97/16784である。もう1件は国際出願番号PCT/JP97/01492である。)。これは、LYEE (Governmental Methodology for Software Providence) と呼ばれるものである。

【0010】

LYEEの定理は、" ソフト要件を同期化させる唯一とつの論理関数をコンピュータ空間に成立させることが出来る。その論理関数を T_0 とすればコンピュータ空間は T_0^n ($n \geq 1$) で表される。 T_0 をシナリオ関数と呼ぶ。" というものである。

【0011】

ここで、シナリオ関数は、

$$T_0 = \Phi_0 \left(\left\{ \Phi_p \{L_{E, i, W02}\} E \right\} e \right. \\ \left. + \Phi_p \{Y_{D, i, W03} \cdot L_{D, i, W03}\} d \right. \\ \left. + \left\{ \Phi_p \{Y_{E, i, W04} \cdot L_{E, i, W04}\} E \right\} e \right)$$

ここで

$$\Phi_p \{L_{E, i, W02}\} E = W02 \text{ パレット}$$

$$\Phi_p \{Y_{D, i, W03} \cdot L_{D, i, W03}\} d = W03 \text{ パレット}$$

$$\Phi_p \{Y_{E, i, W04} \cdot L_{E, i, W04}\} E = W04 \text{ パレット}$$

E ; 画面や電文等イベント情報からなる定義体の識別子という独立変数である。

e ; 開発システムに存在するEの種類という独立変数である。

【0012】

D ; 画面、電文、ファイル等定義体の識別子という独立変数である (EはDに含まれる。)。

【0013】

d ; 開発システムに存在するDの種類という独立変数である (eはdに含まれる

。)。 i ; E や D に含まれる単語（データアイテム）の識別子という独立変数である。 Y ; 位相要素と称し、意味領域間の同期を図る関数である。

【0014】

L ; 論理要素と称し、記憶が存在している場合は記憶との同期、記憶が存在していない場合は演算による同期を図る関数である。その論理構造は図4～図7に示すように一義的である。

【0015】

$\Phi_p \{ \}$; パレット関数と称し、位相要素 Y_{D, i, W_n} 及び論理要素 L_{D, i, W_n} を集合化する作用の関数である。集合化された位相要素、論理要素及びデータエリア D, i, W_n をパレットという。

【0016】

$W02, W03, W04$; 意味領域（パレット）識別子である（意味領域発生順位は、 $W04 \rightarrow W02 \rightarrow W03$ で固定。）。

【0017】

Φ_0 ; パレット連鎖関数と称し、パレットを連鎖させる作用の関数である。パレットの連鎖順位は、 $W04 \rightarrow W02 \rightarrow W03$ で固定。

【0018】

図4～図7に示した位相要素及び論理要素並びにパレット関数は、図5、図7の論理要素における”データの自己生成”の部分及び図6の論理要素における”経路処理”の部分を除くと、” D, i ”のみが変数である。逆に言うと、”データの自己生成”の部分、”経路処理”の部分及び” D, i ”さえ設定すれば、位相要素及び論理要素並びにパレット関数が規定、つまり位相要素、論理要素及びパレット関数に係るプログラムが決定できる。また、パレット連鎖関数は絶対的に唯一のプログラムである。本発明は、この点に着目したものである。即ち、”データの自己生成”の部分、”経路処理”の部分及び” D, i ”が未定義な位相要素及び論理要素並びにパレット関数を予め用意するとともに、これらを定義するための定義表と位相要素及び論理要素並びにパレット関数における未定義部分との間に予め対応関係を持たせておき、例えば定義表に入力するだけで自動的にソフトウェアを生成するようにしたものである。

【0019】

【発明の実施の形態】

以下、本発明の実施の形態を図面に基づき説明する。

【0020】

まず、LYEEの理論に基づくソフトウェアの構成を説明する。

【0021】

図1はLYEEの理論に基づくソフトウェアの構成を概略的に示した図である。

【0022】

LYEEの理論に基づくソフトウェアは、常に図1に示すような構造、つまり各パレット関数102に各基底論理103が組み込まれた各パレット104がパレット連鎖関数101によって連鎖を起こす構造となっている。

【0023】

パレット連鎖関数101、パレット関数102及び基底論理103は、上述したLYEEの理論に基づきそれぞれ1ステップ水準まで画一的に規定されている。

【0024】

基底論理103には、W04基底論理、W02基底論理及びW03基底論理がある。パレット関数102にW04基底論理が組み込まれたパレット104をW04パレット、パレット関数102にW02基底論理が組み込まれたパレット104をW02パレット及びパレット関数102にW03基底論理が組み込まれたパレット104をW03パレットと呼ぶ。各パレットには、図1に示したように、各単語毎にデータを保持するデータエリアが組み込まれている。W04パレットのデータエリアをW04データエリア、W02パレットのデータエリアをW02データエリア及びW03パレットのデータエリアをW03データエリアと呼ぶ。

W04基底論理には、位相要素と論理要素とがある。これらをW04位相要素、W04論理要素と呼ぶ。W02基底論理には、論理要素がある。これをW02論理要素と呼ぶ。W03基底論理には、位相要素と論理要素とがある。これらをW03位相要素、W03論理要素と呼ぶ。

【0025】

W04位相要素及びW04論理要素は、画面編集を行うためのプログラム要素である。W02論理要素は、処理経路を決定するためのプログラム要素である。W03位相要素及びW03論理要素は、ファイル更新のデータ編集を行うためのプログラム要素である。

【0026】

これらの基底論理103は、生産するソフトウェアに必要な画面105（画面の他ファイルまたは帳票も含む。）に存在する全ての単語に対して作成される。図1の例で言うと、画面105上に例えば単語「暗証番号」、「顧客氏名」、「預金残高」、「メッセージ」、「実行」、「終了」なる単語が存在するとすると、これらの各単語に対して基底論理が作成される。なお、基底論理のうち、W04位相要素、W04論理要素及びW2論理要素はそれぞれ画面編集を行うためのもの、処理経路を決定するためのものであるから画面毎に定義されるものであるが、W03位相要素及びW03論理要素についてはファイル更新のデータ編集を行うためのものであるからシステム内の定義体全てを括ってなる1個のパレットとして定義される。例えば、「暗証番号」なる単語についてW04位相要素、W04論理要素及びW2論理要素は同じ単語であっても画面毎にそれぞれ別個に定義しなければならない。そして、それぞれ別個に定義された単語がそのまま個別にW03論理要素として定義すればよい。

【0027】

これら各基底論理には、意味領域識別子、定義体識別子及び単語識別子が付される。

【0028】

図2はパレット連鎖関数101の構造パラダイムを示すフローチャートである。

【0029】

パレット連鎖関数101の構造はいかなるソフトウェアにおいても同一である。従って、予め作成してあるパレット連鎖関数101に作成したパレット104を組み込めばよい。

【0030】

即ち、まず送信情報を決定する（ステップ201）。送信情報とは、コンピュータ上で表示すべき画面105の表示データの事である。

【0031】

例えば図11に示す処理経路図（処理経路図の詳細については後述する。）を参照しながら説明すると、例えばメニュー画面GSMENU（GSMENUはメニュー画面の定義体識別子）において「残高照会」が選択されると（例えばメニュー画面GSMENUにおいて「メニューNo=n+ENTER」の操作）、「残高照会」の画面GSOI08（GSOI08は残高照会画面の定義体識別子）を送信情報と決定する。

【0032】

送信情報にシステムを閉鎖する情報（メニュー画面GSMENU「終了」）が含まれているような場合には（ステップ202）、システムを閉鎖する（ステップ203）。システムを閉鎖するとは、例えばOSに戻る事である。

【0033】

次に、WT単位内の継続かどうかを判断する（ステップ204）。これは例外的処理に関するものである。即ち、WT単位とは、例えばデータの登録に使用される登録画面と当該データの登録の際のデータの参照に使用される参照画面とを1つの単位で括ってなるものである。WT単位内の継続でない場合には、新たなWT単位の全パレットをセットする（ステップ207）。

【0034】

次に、該当するW04パレットを実行する（ステップ208）。即ち、パレット関数102に全てのW04に関する基底論理103つまりW04位相要素とW04論理要素がセットされたW04パレットを実行する。これにより表示すべき画面に関するデータ値が決定される。

【0035】

そして、送信機能を実行する（ステップ209）。即ち、データ値等がセットされた画面をコンピュータ上で表示する画面データとして送信する。

【0036】

その後、受信機能を実行する（ステップ210）。即ち、コンピュータの画面上でデータ値等が入力された画面データを受信する。

【0037】

そして、受信データが正常かどうかを判断し（ステップ211）、異常の場合には最初からやり直す。即ち、規則違反のデータ等がないかを判断する。

【0038】

次に、該当するW02パレットを実行する（ステップ212）。即ち、パレット関数102に全てのW02に関する基底論理つまりW02論理要素がセットされたW02パレットを実行する。これによりコンピュータの画面上で入力されたデータ等が決定される。

【0039】

次に、処理経路が決定される（ステップ213、214）。処理経路はW02論理要素上の処理経路フラグ（R=1～5）によって判断される。

【0040】

処理経路フラグがR=1の場合には、該当するW03パレット（ファイルへの登録を伴わないW03パレット）を実行する（ステップ215）。そして、最初の処理（ステップ201）に戻る。処理経路R=1はデータ処理の実行のみを行う経路処理である。

【0041】

処理経路フラグがR=2の場合には、該当するW03パレット（ファイルへの登録を伴うW03パレット）を実行する（ステップ216）。そして、最初の処理（ステップ201）に戻る。処理経路R=2はデータ処理の実行とデータのファイルへの登録を行う経路処理である。

【0042】

処理経路フラグがR=3～5の場合には、そのまま最初の処理（ステップ201）に戻る。

【0043】

ここで、処理経路R=3はそのままW04パレット（同一、異種の両方）に戻

る経路処理である。

【0044】

経路処理 R = 4 はデータをそのままファイルに登録する処理である。

【0045】

処理経路 R = 5 はデータをそのままファイルから取り出す処理である。

【0046】

図3はパレット関数の構造パラダイムを示すフローチャートである。

【0047】

W02、W03、W04に関する基底論理それぞれについてこのような構造のパレット関数を作成する必要がある。

【0048】

まずデータエリアを“空”化する（ステップ301）。即ち、例えばW04に関するパレット関数では画面ごとに、当該画面に存在する単語毎にその識別子が付されたデータエリアを“空”化する。ただし、W02上のパレット関数だけは「空化」を行わない。そして、各位相要素と論理要素を順次（位相要素、論理要素の順番で）実行する（ステップ302、303）。

【0049】

その後、パレット再起動フラグがセットされている場合には（ステップ304）、フラグをリセットする（ステップ305）と共に各位相要素と論理要素を順次（位相要素、論理要素の順番で）再実行する（ステップ302、303）。

一方、パレット再起動フラグがセットされていない場合にはファイルを閉じて（ステップ306）処理を終了する。ここで、「ファイル」とは、W03の基底論理で自己生成されたデータ値を「記憶」するものであって、単語毎に設けられる。なお、ファイルに記憶されたデータの再現は、単語毎に再現元ファイルが異なるので、単語毎の論理要素の自己生成部分のサブルーチンで行う。そして、サブルーチン内でファイルのオープンとリードが実行され、クローズはパレット関数が纏めて実行する。

【0050】

パレット再起動は、各論理要素上の再起動フラグによって判断される。

【0051】

図4は基底論理のうち位相要素の構造パラダイムを示すフローチャートである。

【0052】

位相要素を $Y(D, i, W_n)$ とすると、まず D, i, W_{n-1} (D は画面識別子、 i は単語識別子、 W_{n-1} は例えば位相要素が $W03$ なら $W02$ を示す。)のデータエリアが空でないか否かを判別する(ステップ401)。

【0053】

空の場合には位相要素 $Y(D, i, W_n)$ を終了する。

【0054】

空でない場合には、 D, i, W_{n-1} のデータエリアのデータを D, i, W_n のデータエリアに移して終了する(ステップ402)。

【0055】

図5は基底論理のうち $W03$ 論理要素の構造パラダイムを示すフローチャートである。

【0056】

論理要素を $L(D, i, W03)$ とすると、まずデータエリア($D, i, W03$)が空でないか否かを判別する(ステップ501)。

【0057】

データエリア($D, i, W03$)に既にデータがあって空でない場合には、論理要素 $L(D, i, W03)$ を終了する。

【0058】

データエリア($D, i, W03$)にデータがなくて空の場合には、データの自己生成を行う(ステップ502)。データの自己生成とは、データが入力属性の場合には例えばデータエリア($D, i, W02$)のデータをデータエリア($D, i, W03$)に移すこと、データが出力属性の場合には例えばファイルの同一単語のデータからデータを自己生成すること、データが同様に出力属性の場合には例えば他の単語のデータから演算でデータを自己生成すること、をいう。

【0059】

次に、データエリア（D、i、W03）が空でないか否かを判別する（ステップ503）。

【0060】

データエリア（D、i、W03）にデータが存在することとなった場合には、自己生成が成立したとみなし、論理要素L（D、i、W03）を終了する。

データエリア（D、i、W03）にデータが存在しない場合には、D、i、W03の再起動カウンタに”1”を加算する（ステップ504）。

【0061】

次に、再起動カウンタの値が予め定めた値以下かどうか、即ち再起動可能かどうかを判定する（ステップ505）。

【0062】

再起動可能な場合には、再起動フラグ（D、i、W03）をセットする（ステップ506）。

【0063】

再起動可能でない場合には、不成立フラグ（D、i、W03）をセットする（ステップ507）。

【0064】

図6は基底論理のうちW02論理要素の構造パラダイムを示すフローチャートである。論理要素をLとすると、まずデータエリア（D、i、W02）が空でないか判別する（ステップ601）。

【0065】

データエリア（D、i、W02）にデータがなくて空の場合、つまり画面からデータの入力が行われなかった場合には、論理要素L（D、i、W02）を終了する。

【0066】

データエリア（D、i、W02）に既にデータがあって空でない場合には、処理経路を設定する（ステップ602）。処理経路の設定とは、上述した処理経路フラグ（R=1～5）を設定することである。

【0067】

次に、処理経路の設定が成立したか否かを判別する（ステップ603）。

【0068】

処理経路の設定が成立した場合には、論理要素L（D、i、W02）を終了する。

【0069】

処理経路の設定が成立しない場合には、D、i、W02の再起動カウンタに“1”を加算する（ステップ604）。

【0070】

次に、再起動カウンタの値が予め定めた値以下かどうか、即ち再起動可能かどうかを判定する（ステップ605）。

【0071】

再起動可能な場合には、再起動フラグ（D、i、W02）をセットする（ステップ606）。

【0072】

再起動可能でない場合には、不成立フラグ（D、i、W02）をセットする（ステップ607）。

【0073】

図7は基底論理のうちW04論理要素の構造パラダイムを示すフローチャートである。

【0074】

論理要素をL（D、i、W04）とすると、まずデータエリア（D、i、W04）が空でないか否かを判別する（ステップ701）。

【0075】

データエリア（D、i、W04）にデータがなくて空の場合には、データの自己生成を行う（ステップ702）。データの自己生成とは、データが入力属性の場合には例えばデータエリア（D、i、W02）のデータをデータエリア（D、i、W04）に移すこと、データが出力属性の場合には例えばファイルの同一単語のデータからデータを自己生成すること、データが同様に出力属性の場合には

例えば他の単語のデータから演算でデータを自己生成すること、をいう。

【0076】

次に、データエリア(D、i、W04)が空でないか否かを判別する(ステップ703)。

【0077】

データエリア(D、i、W04)にデータが存在することとなった場合には、自己生成が成立したとみなし、論理要素L(D、i、W04)を終了する。

【0078】

データエリア(D、i、W04)にデータがなくて“空”の場合には、D、i、W04の再起動カウンタに“1”を加算する(ステップ704)。

【0079】

次に、再起動カウンタの値が予め定めた値以下かどうか、即ち再起動可能かどうかを判定する(ステップ705)。

【0080】

再起動可能な場合には、再起動フラグ(D、i、W04)をセットする(ステップ706)。

【0081】

再起動可能でない場合には、不成立コードをセットする(ステップ707)。

一方、ステップ701において、データエリア(D、i、W04)に既にデータがあって空でない場合には、該単語のW02不成立フラグ又は該単語のW03不成立フラグがセットされているかどうかを判断する(ステップ708)。

【0082】

単語のW02不成立フラグ又は該単語のW03不成立フラグがセットされている場合には、不成立コードをセットするとともに、不成立フラグをリセットする(ステップ709)。

【0083】

なお、不成立コードは、例えば図10に示すように、「暗証番号」のデータエリア1001に誤った番号が入力されたときに「メッセージ」のデータエリアにその旨を表示するために使われる。「暗証番号」のW04論理要素で不成立コ

ードがセットされると、「メッセージ」のW04論理要素の「データの自己生成」によってその旨が表示される。

【0084】

本発明は図3に示したパレット関数、図4～図7に示した位相要素及び論理要素に係るプログラムを自動的に生成するものである。パレット連鎖関数は常に同一のプログラムで構成できるので、パレット関数、位相要素及び論理要素に係るプログラムを自動的に生成できるということは、目的とするソフトウェアを自動的に生成することを意味する。そして、パレット関数、位相要素及び論理要素のいずれもデータエリアのアドレスだけでロジックを実現していることから、業務用ソフトウェアやゲーム用ソフトウェア、その他あらゆる分野のソフトウェアを自動的に生成できることを意味する。

【0085】

図8は本発明に係るソフトウェアの自動生成装置の構成を示すブロック図である。このソフトウェアの自動生成装置は、コンピュータ上でソフトウェアにより実現することができる。

【0086】

図8において、801は図4～図7に示した位相要素、論理要素及びパレット関数に係るプログラムであって”データの自己生成”の部分、”処理経路”の部分及び”D、i”が未定義なプログラムを予め記憶する第1の記憶部、802はパレット連鎖関数に係るプログラムを予め記憶する第2の記憶部、803は”データの自己生成”の部分、”処理経路”の部分及び”D、i”を入力するための入力部である。

【0087】

また、804は入力部803で入力された”データの自己生成”の部分、”処理経路”の部分及び”D、i”を第1の記憶部802に記憶された位相要素、論理要素及びパレット関数に係るプログラムの所定の位置に埋め込む埋込部、805は定義後の位相要素、論理要素及びパレット関数に係るプログラムをパレット連鎖関数に組み込む組込部、806はプログラムの組み込まれたソフトウェアを出力する出力部である。

【0088】

次に、このように構成された装置を使ってソフトウェアを生成する手順を図9に沿って説明する。

【0089】

まず、目的とするソフトウェアで使用する画面や帳票を特定し（ステップ901）、それぞれの画面に識別子（定義体識別子）を設定する（ステップ902）。図10は決定された画面の一例である。図10の画面にシステムユニークな任意の識別子、例えば“GSOI08”といった定義体識別子を設定する。また、図10において右側の空欄はデータフィールドを示している。

【0090】

次に、処理経路図を作成し、ファイルを決める（ステップ903）。処理経路図は決定された画面や帳票に基づき作成される。処理経路図は人間が作成しても良いし、機械化しても良い。処理経路図は各画面や各帳票に対応するパレット（ボックスで表示）を線で接続したものである。線はパレット連鎖関数を意味する。画面や帳票が決定されていればパレット間の接続は必然的に決まる。図11は処理経路図の一例を示している。

【0091】

処理経路図においては、例えばある画面（例えば「メニュー画面GSMENU」）が決まるとその画面での処理内容（「メニューNO=n+ENTER」、「終了」の画面上の制御単語）から次に必要な処理（例えば「メニュー画面GSMENU」からは「残高照会画面GSOI08」の表示や「システム閉鎖」、
「残高照会画面GSOI08」からは「実行」の制御単語によってファイルFSHINFILからのデータ（残高データ）の読み出しや暗証番号エラーに対する元の画面への復帰あるいは「終了」の制御単語によってメニュー画面GSMENUへの復帰。）が必然的に決まるので、上記処理内容に応じてパレット等の間を線で結んでいる。

【0092】

図11において、「GSMENU」の下に「W04」は画面識別子GSMENUが付されたW04パレットであり、このW04パレットによってメニュー画面

GSMENUが形成される。

【0093】

「GSMENU」の下「W02」は画面識別子GSMENUが付されたW02パレットであり、このW02パレットによって処理経路が決定される。

【0094】

「GSOI08」の下「W04」は画面識別子GSOI08が付されたW04パレットであり、このW04パレットによって残高照会画面GSOI08のデータフィールドに表示するデータが形成される。

【0095】

「GSOI08」の下「W02」は画面識別子GSOI08が付されたW02パレットであり、このW02パレットによって処理経路が決定される。

【0096】

「W03」はシステムに一個のW03パレットであり、このW03パレットによって預金残高が算出される。算出された預金残高はW04パレットの位相要素、論理要素の実行を経て残高照会画面GSOI08に表示される。

【0097】

作成された処理経路図に基づき必要なファイルを決定する。即ち必要なファイルは処理経路図より必然的に決定できる（例えば図11では「FSHINFIL」「FMOTOFIL」。「FSHINFIL」は、「顧客台帳ファイル」の定義体識別子を意味し、「FMOTOFIL」は、「預金元帳ファイル」の定義体識別子を意味する。）。決定したファイルは処理経路図に記述する。

【0098】

なお、「IO-W04-RK1-SEC」とは、「顧客台帳ファイル」からデータをリードするW04パレット上のサブルーチン識別子である。「IO-W04-RD1-SEC」とは、「預金元帳ファイル」からデータをリードするW04パレット上のサブルーチン識別子である。「IO-W02-RK1-SEC」とは、「顧客台帳ファイル」からデータをリードするW02パレット上のサブルーチン識別子である。「IO-W03-RK1-SEC」とは、「顧客台帳ファイル」からデータをリードするW03パレット上のサブルーチン識別子である。

「I O-W 0 3 R D 1-S E C」とは、「預金元帳ファイル」からデータをリードするW 0 3 パレット上のサブルーチン識別子である。以上は図1においても同様である。

【0099】

次に、特定された画面、帳票、ファイルから単語を収集し（ステップ904）、それぞれの単語に識別子を設定する（ステップ905）。例えば図10の画面から「暗唱番号」、「顧客氏名」、「預金残高」、「メッセージ」、「実行」、「終了」なる単語を収集し、例えば「暗唱番号」、「顧客氏名」、「預金残高」、「メッセージ」、「実行」、「終了」に対して単語識別子として"PASSN 01", "CLNAME", "YOKNZAN", "MSSGCD1", "PF25KEY", "PF3KEY"を設定する。

【0100】

図12は入力部803より入力される定義表の一例を示す図である。この定義表は例えばコンピュータ画面上に表示され、そこから各データが入力される。

【0101】

定義表では、画面識別子（I D）の入力欄、画面名称の入力欄、単語名称の入力欄がある。また、各単語名称に対応して、単語識別子（I D）、データが入力属性か出力属性か（I / O）、データの桁数、データ属性が数字型か文字型（属性）、使用ファイル（上述したように記憶されたデータの再現は単語毎に再現元ファイルが異なるので単語毎の論理要素の自己生成部分のサブルーチンで行う。そのために単語毎に使用ファイルを定義している。）、処理経路（R-N O）、処理経路決定単語に対応した機能キー番号（F-K E Y）、次画面識別子、単語定義（データの自己生成）の入力項目がある。単語定義（データの自己生成）の入力項目には、データが入力属性の場合にはL Y E Eの原理により自動的に決定できるので何も定義しない。データが出力属性の場合、例えばファイルの同一単語のデータからデータを自己生成するか、データが同様に出力属性の場合には例えば他の単語のデータから演算でデータを自己生成するかどうか、を設定する。

より具体的には、「ファイルの同一単語のデータからデータを自己生成」とは、「顧客氏名」、「預金残高」がその一例である。

【0102】

また、「他の単語のデータから演算でデータを自己生成」とは、「預金残高」(＝ファイルの預金残高＋画面の取引金額)がその一例である。

【0103】

処理経路の入力項目には、上述した処理経路(R＝1～5)を設定する。

【0104】

本発明でいう「導出データ」は、上記の単語定義に相当する。

【0105】

また、本発明でいう「処理経路」は、上記の処理経路(R-NO)、機能キー番号(F-KEY)及び次画面識別子に相当する。

【0106】

そして、埋込部804及び組込部805によってソフトウェアの自動生成を実行する(ステップ907)。

【0107】

この自動生成の動作を図13のフローチャートに基づき説明する。

【0108】

即ち、定義表に画面識別子や単語識別子等、つまり本発明でいう”データの自己生成”の部分、”処理経路”の部分及び”D、i”が入力され(ステップ1301)、自動生成の実行がなされると(ステップ1302)、埋込部804が第1の記憶部801から”データの自己生成”の部分、”処理経路”の部分及び”D、i”が未定義の位相要素、論理要素及びパレット関数を呼び込み(ステップ1303)、未定義の位相要素、論理要素及びパレット関数の所定の位置に定義表に設定された”データの自己生成”の部分、”処理経路”の部分及び”D、i”がそれぞれ埋め込まれる(ステップ1304)。

【0109】

この点をさらに具体的に説明する。

【0110】

未定義のW03論理要素のプログラムは以下の通りである。

【0111】

*-----

* @%8@

*-----

L-W03-@%16@-@%19@-SEC SECTION.

L-W03-@%16@-@%19@-START.

*--< 有意判定 >--

IF @%16@-@%19@ OF W03 NOT = @%20@

GO TO L-W03-@%16@-@%19@-EXIT

END-IF.

*--< 自己生成 >--

*--< 入力属性のとき W02 をセット >--

* MOVE @%16@-@%19@ OF W02

* TO @%16@-@%19@ OF W03.

*--< 出力属性のとき、ファイルに同一単語があればそこから自己生成 >

* MOVE G#####-***** OF W03

* TO F#####-***** OF F#####-REC.

* PERFORM IO-W03-RD*-SEC.

* IF RD*-INVS = "IV"

* MOVE @%20@ TO @%16@-@%19@ OF W03

* ELSE

* MOVE F#####-@%19@ OF F#####-REC

* TO @%16@-@%19@ OF W03

* END-IF.

*--< 出力属性のとき 他の単語との演算で自己生成 >--

* COMPUTE @%16@-@%19@ OF W03

* = ###-### OF W03

* + ***-*** OF W03.

*--< 成立判定 >--

```

IF      @%16@-@%19@  OF W03  = @%20@
*      OR  @%16@-@%19@  OF W03  > 0
      ADD 1    TO  @%16@-@%19@-CNT
      IF  @%16@-@%19@-CNT  <  RECALL-MAX
      MOVE "1"  TO  RECALL-FLG
      ELSE
      MOVE "1"  TO  @%16@-@%19@-NON
      END-IF
      END-IF.
      L-W03-@%16@-@%19@-EXIT.
      EXIT. 」

```

この未定義のW03論理要素のプログラムにおいて、

@%8@は単語名称、

@%16@は画面識別子、

@%19@は単語識別子、

@%20@は属性に対応した「空」の値、

F#####はファイルアクセスのキーとなる定義体識別子

G#####はファイルアクセスのキーのもととなる単語が存在する画面の定義体識別子、

RD*はアクセスするファイルに対応したサブルーチンに固有する識別子、

****はファイルアクセスのキーとなる単語識別子、

###-###及び***-***は演算により導出するための独立変数の単語識別子を表している。

【0112】

そして、図12に示した定義表の

単語名称の欄が@%8@

画面識別子の欄が@16@

単語識別子の欄が@%19@

属性の欄が @ % 2 0 @

単語定義の欄が

*--< 自己生成 >--

*--< 入力属性のとき W 0 2 をセット >--

* MOVE @%16@-@%19@ OF W02

* TO @%16@-@%19@ OF W03.

*--< 出力属性のとき ファイルに同一単語があればそこから自己生成 >

--

* MOVE G#####-***** OF W03

* TO F#####-***** OF F#####-REC.

* PERFORM IO-W03-RD*-SEC.

* IF RD*-INVS = "IV"

* MOVE @%20@ TO @%16@-@%19@ OF W03

* ELSE

* MOVE F#####-@%19@ OF F#####-REC

* TO @%16@-@%19@ OF W03

* END-IF.

*--< 出力属性のとき 他の単語との演算で自己生成 >--

* COMPUTE @%16@-@%19@ OF W03

* = ###-### OF W03

* + ***-*** OF W03.

に対応している。

【 0 1 1 3 】

このうち

*--< 入力属性のとき W 0 2 をセット >--

* MOVE @%16@-@%19@ OF W02

* TO @%16@-@%19@ OF W03.

がデータエリア (D、i、W_n) - 1 のデータをデータエリア (D、i、W_n)

に移すプログラム部分、

*--< 出力属性のとき ファイルに同一単語があればそこから自己生成 >

--

```
*      MOVE G#####-***** OF  W03
*      TO  F#####-***** OF  F#####-REC.
*      PERFORM  IO-W03-RD*-SEC.
*      IF  RD*-INVS  =  "IV"
*      MOVE  @%20@      TO  @%16@-@%19@  OF  W03
*      ELSE
*      MOVE  F#####-@%19@ OF  F#####-REC
*      TO  @%16@-@%19@  OF  W03
*      END-IF.
```

がファイルの同一単語のデータからデータを自己生成するプログラム部分、

*--< 出力属性のとき 他の単語との演算で自己生成 >--

```
*      COMPUTE  @%16@-@%19@  OF  W03
*
*      = ###-### OF  W03
*
*      + ***-*** OF  W03.
```

が他の単語のデータから演算でデータを自己生成するプログラム部分を示している。

【0114】

そして、定義表に画面識別子等が設定されると、単語ごとに未定義の論理要素の各欄、@%8@、@%16@、@%19@、@%20@、#####、###、*、*****、###-###、***-***にその値が設定され、以下に示す単語毎のW03論理要素、つまり完成されたW03論理要素のプログラムが生成される。

【0115】

*

*-----

* 暗証番号

*-----

L-W03-GSOI08-PASSN01-SEC SECTION.

L-W03-GSOI08-PASSN01-START.

*--< 有意判定 >--

IF GSOI08-PASSN01 OF W03 NOT = LOW-VALUE

GO TO L-W03-GSOI08-PASSN01-EXIT

END-IF.

*--< 自己生成 >--

*--< 入力属性のとき W02 をセット >--

MOVE GSOI08-PASSN01 OF W02

TO GSOI08-PASSN01 OF W03

*--< 成立判定 >--

IF GSOI08-PASSN01 OF W03 = LOW-VALUE

OR GSOI08-PASSN01 OF W03 > 0

ADD 1 TO GSOI08-PASSN01-CNT

IF GSOI08-PASSN01-CNT < RECALL-MAX

MOVE "1" TO RECALL-FLG

ELSE

MOVE "1" TO GSOI08-PASSN01-NON

END-IF

END-IF.

L-W03-GSOI08-PASSN01-EXIT.

EXIT.

*

*-----

* 顧客氏名

*-----

L-W03-GS0108-CLNAME-SEC SECTION.

L-W03-GS0108-CLNAME-START.

*--< 有意判定 >--

IF GS0108-CLNAME-XX OF W03 NOT = ALL "9"

GO TO L-W03-GS0108-CLNAME-EXIT

END-IF.

*--< 自己生成 >--

*--< 出力属性のとき ファイルに同一単語があればそこから自己生成 >

MOVE GS0108-PASSNO1 OF W03

TO FMOTOFIL-PASSNO1 OF FMOTOFIL-REC.

PERFORM IO-W03-RD1-SEC.

IF RD1-INVSN = "IV"

MOVE ALL "9" TO GS0108-CLNAME-XX OF W03

ELSE

MOVE FMOTOFIL-CLNAME OF FMOTOFIL-REC.

TO GS0108-CLNAME OF W03

END-IF.

*--< 成立判定 >--

IF GS0108-CLNAME-XX OF W03 = ALL "9"

OR GS0108-CLNAME-XX OF W03 > 0

ADD 1 TO GS0108-CLNAME-CNT

IF GS0108-CLNAME-CNT < RECALL-MAX

MOVE "1" TO RECALL-FLG

ELSE

MOVE "1" TO GS0108-CLNAME-NON

END-IF

END-IF.

L-W03-GS0108-CLNAME-EXIT.

EXIT.

*

*-----

* 預金残高

*-----

L-W03-GS0108-YOKNZAN-SEC SECTION.

L-W03-GS0108-YOKNZAN-START.

*--< 有意判定 >--

IF GS0108-YOKNZAN OF W03 NOT = LOW-VALUE

GO TO L-W03-GS0108-YOKNZAN-EXIT

END-IF.

*--< 自己生成 >--

*--< 出力属性のとき ファイルに同一単語があればそこから自己生成 >

MOVE G S0108- PASSNO1 OF W03

TO FMOTOFIL- PASSNO1 OF FMOTOFIL-REC.

PERFORM IO-W03-RD1-SEC.

IF RD1-INVS = "IV"

MOVE ALL "9" TO GS0108-CLNAME-XX OF W03

ELSE

MOVE FMOTOFIL-YOKNZAN OF FMOTOFIL-REC

TO GS0108-YOKNZAN OF W03

END-IF.

*--< 成立判定 >--

IF GS0108-YOKNZAN OF W03 = LOW-VALUE

OR GS0108-YOKNZAN OF W03 > 0

ADD 1 TO GS0108-YOKNZAN-CNT

```

IF  GSOI08-YOKNZAN-CNT <  RECALL-MAX
    MOVE  "1"  TO  RECALL-FLG
ELSE
    MOVE  "1"  TO  GSOI08-YOKNZAN-NON
END-IF

```

END-IF.

L-W03-GSOI08-YOKNZAN-EXIT.

EXIT.

*

*-----

* メッセージ

*-----

L-W03-GSOI08-MSSGCD1-SEC SECTION.

L-W03-GSOI08-MSSGCD1-START.

*--< 有意判定 >--

```

IF  GSOI08-MSSGCD1  OF W03  NOT = LOW-VALUE
    GO  TO  L-W03-GSOI08-MSSGCD1-EXIT

```

END-IF.

*--< 自己生成 >--

*--<自己生成なし

*--< 成立判定 >--

```

IF  GSOI08-MSSGCD1  OF W03  = LOW-VALUE
OR  GSOI08-MSSGCD1  OF W03  > 0
    ADD  1  TO  GSOI08-MSSGCD1-CNT
    IF  GSOI08-MSSGCD1-CNT <  RECALL-MAX
        MOVE  "1"  TO  RECALL-FLG
    ELSE
        MOVE  "1"  TO  GSOI08-MSSGCD1-NON
END-IF

```

END-IF.

L-W03-GS0108-MSSGCD1-EXIT.

EXIT.

*

*-----

* 終了

*-----

L-W03-GS0108-PF3KEY-SEC SECTION.

L-W03-GS0108-PF3KEY-START.

*--< 有意判定 >--

IF GS0108-PF3KEY OF W03 NOT = LOW-VALUE

GO TO L-W03-GS0108-PF3KEY-EXIT

END-IF.

*--< 自己生成 >--

*--< 自己生成なし

*--< 成立判定 >--

IF GS0108-PF3KEY OF W03 = LOW-VALUE

OR GS0108-PF3KEY OF W03 > 0

ADD 1 TO GS0108-PF3KEY-CNT

IF GS0108-PF3KEY-CNT < RECALL-MAX

MOVE "1" TO RECALL-FLG

ELSE

MOVE "1" TO GS0108-PF3KEY-NON

END-IF

END-IF.

L-W03-GS0108-PF3KEY-EXIT.

EXIT.

*

*-----

* 実行

*-----

L-W03-GS0108-PF25KEY-SEC SECTION.

L-W03-GS0108-PF25KEY-START.

*--< 有意判定 >--

IF GS0108-PF25KEY OF W03 NOT = LOW-VALUE

GO TO L-W03-GS0108-PF25KEY-EXIT

END-IF.

*--< 自己生成 >--

*--< 自己生成なし

*--< 成立判定 >--

IF GS0108-PF25KEY OF W03 = LOW-VALUE

OR GS0108-PF25KEY OF W03 > 0

ADD 1 TO GS0108-PF25KEY-CNT

IF GS0108-PF25KEY-CNT < RECALL-MAX

MOVE "1" TO RECALL-FLG

ELSE

MOVE "1" TO GS0108-PF25KEY-NON

END-IF

END-IF.

L-W03-GS0108-PF25KEY-EXIT.

EXIT.

また、未定義のW03位相要素は、以下の通りである。

【0116】

*

*-----

* @%8@

*-----

Y-W03-@%16@-@%19@-SEC SECTION.

Y-W03-@%16@-@%19@-START.

IF @%16@-@%19@ OF W02 NOT = @%20@

MOVE @%16@-@%19@ OF W02

TO @%16@-@%19@ OF W03

END-IF.

Y-W03-@%16@-@%19@-EXIT.

EXIT.

そして、単語ごとに未定義のW 0 3 位相要素の各欄にその値が設定され、以下に示す単語毎のW 0 3 位相要素、つまり完成されたW 0 3 位相要素のプログラムが生成される。

【 0 1 1 7 】

*-----
* 暗証番号
*-----

Y-W03-GS0I08-PASSN01-SEC SECTION.

Y-W03-GS0I08-PASSN01-START.

IF GS0I08-PASSN01 OF W02 NOT = LOW-VALUE

MOVE GS0I08-PASSN01 OF W02

TO GS0I08-PASSN01 OF W03

END-IF.

Y-W03-GS0I08-PASSN01-EXIT.

EXIT.

*-----
* 顧客氏名
*-----

Y-W03-GSOI08-CLNAME-SEC SECTION.

Y-W03-GSOI08-CLNAME-START.

IF GSOI08-CLNAME-XX OF W02 NOT = ALL "9"

MOVE GSOI08-CLNAME OF W02

TO GSOI08-CLNAME OF W03

END-IF.

Y-W03-GSOI08-CLNAME-EXIT.

EXIT.

*

*-----

* 預金残高

*-----

Y-W03-GSOI08-YOKNZAN-SEC SECTION.

Y-W03-GSOI08-YOKNZAN-START.

IF GSOI08-YOKNZAN OF W02 NOT = LOW-VALUE

MOVE GSOI08-YOKNZAN OF W02

TO GSOI08-YOKNZAN OF W03

END-IF.

Y-W03-GSOI08-YOKNZAN-EXIT.

EXIT.

*

*-----

* メッセージ

*-----

Y-W03-GSOI08-MSSGCD1-SEC SECTION.

Y-W03-GSOI08-MSSGCD1-START.

IF GSOI08-MSSGCD1 OF W02 NOT = LOW-VALUE

MOVE GSOI08-MSSGCD1 OF W02

TO GSOI08-MSSGCD1 OF W03

END-IF.

Y-W03-GS0108-MSSGCD1-EXIT.

EXIT.

*

*-----

* 終了

*-----

Y-W03-GS0108-PF3KEY-SEC SECTION.

Y-W03-GS0108-PF3KEY-START.

IF GS0108-PF3KEY OF W02 NOT = LOW-VALUE

MOVE GS0108-PF3KEY OF W02

TO GS0108-PF3KEY OF W03

END-IF.

Y-W03-GS0108-PF3KEY-EXIT.

EXIT.

*

*-----

* 実行

*-----

Y-W03-GS0108-PF25KEY-SEC SECTION.

Y-W03-GS0108-PF25KEY-START.

IF GS0108-PF25KEY OF W02 NOT = LOW-VALUE

MOVE GS0108-PF25KEY OF W02

TO GS0108-PF25KEY OF W03

END-IF.

Y-W03-GS0108-PF25KEY-EXIT.

EXIT.

未定義のW03論理要素群は以下の通りである。

【0118】

*--< @%8@

PERFORM L-W03-@%16@-@%19@-SEC.

未定義のW03位相要素群は以下の通りである。

【0119】

*

*--< @%8@

PERFORM Y-W03-@%16@-@%19@-SEC.

また、未定義のパレット関数は以下の通りである。

【0120】

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SOURCE-COMPUTER. AS400.

OBJECT-COMPUTER. AS400.

INPUT-OUTPUT SECTION.

FILE-CONTROL.

*

-- KEY有り--

SELECT @%16@ ASSIGN TO DATABASE-@%16@

ORGANIZATION IS INDEXED

ACCESS MODE IS DYNAMIC

RECORD KEY IS EXTERNALLY-DESCRIBED-KEY

* WITH DUPLICATES

FILE STATUS IS @%24@.

-- KEY無し--

```
*      SELECT @%16@      ASSIGN      TO DATABASE-@%16@
*
*      ORGANIZATION IS SEQUENTIAL
*
*      ACCESS  MODE IS SEQUENTIAL
*
*      FILE  STATUS IS @%24@.
```

DATA DIVISION.

FILE SECTION.

FD @%16@

LABEL RECORD IS STANDARD.

01 @%16@-R.

COPY DDS-ALL-FORMATS OF @%16@.

01 @%16@-REC.

COPY @%9@.

WORKING-STORAGE SECTION.

01 PALETTE-WORK.

03 PS PIC S9(03) COMP.

01 WK-FILE-STATUS.

03 @%24@ PIC X(02).

01 FILE-STATUS.

03 @%7@-OPNS PIC X(01).

03 @%7@-REDS PIC X(01).

03 @%7@-INVS PIC X(02).

*--<W 0 3 画面クリアに使用する。

【 0 1 2 1 】

01 WK-@%16@.

COPY @%16@.

LINKAGE SECTION.

--< CTRL >--

01 CTRL-AREA.

COPY CTRLAREA.

--< W02 >--

01 W02-AREA.

02 W02 OCCURS 5.

COPY W0204.

* --< 画面 >--

03 W02-@%16@ REDEFINES INFO.

COPY @%16@.

03 @%16@-NON REDEFINES NON-FLG.

COPY @%16@N.

--< W03 >--

01 W03-AREA.

02 W03.

* --< I / F 情報 >--

03 W03-CNTRL.

05 RECALL-FLG PIC X(1).

05 RECALL-MAX PIC 9(2).

* --< 単語 >--

03 W03-ITEMS.

04 W03-@%16@.

COPY @%16@.

04 W03-FILE.

COPY @%16@.

04 W03-LIST.

COPY @%16@.

* --< 不成立フラグ >--

03 W03-NON.

COPY @%16@N.

COPY @%16@N.

COPY @%16@N.

* --< 再起回数 >--

03 W03-CNT.

COPY @%16@Z.

COPY @%16@Z.

COPY @%16@Z.

--< W04 >--

01 W04-AREA.

02 W04 OCCURS 5.

COPY W0204.

* --< 画面 >--

03 W04-@%16@ REDEFINES INFO.

COPY @%16@.

```
03  OPTION-IDX-DEF  REDEFINES  OPTION-IDX.
```

COPY @%16@C.

*--< WFL >--

01 WFL-AREA.

03 FILLER PIC X(01).

PROCEDURE DIVISION USING CTRL-AREA

W02-AREA

W03-AREA

W04-AREA

WFL-AREA.

* ****

* * * M A I N * *

```

*
*****
MAIN-SEC      SECTION.
MAIN-START.
*--< 前処理 >--
      PERFORM  INIT-PROC.
*--< 位相要素 >--
      PERFORM  Y-PROC.
*--< 論理要素 >--
      PERFORM  UNTIL  RECALL-FLG      =  SPACE
                        MOVE  SPACE      TO  RECALL-FLG
      PERFORM  L-PROC
      END-PERFORM.
*--< Pwrite
      PERFORM  IO-W03-WT*-SEC.
      PERFORM  IO-W03-UP*-SEC.
*--< 後処理 >--
      PERFORM  FIN-PROC.
MAIN-END.
      EXIT PROGRAM.
*
*****
*
*      前処理      *
*
*****
INIT-PROC SECTION.
INIT-START.
*--< 初期値 設定 >--
      INITIALIZE      PALETTE-WORK.
      INITIALIZE      WK-FILE-STATUS.
      INITIALIZE      FILE-STATUS.
      MOVE  SPACE      TO  FLAME-FLG OF CTRL-AREA.

```

```

MOVE 05 TO RECALL-MAX.
MOVE HIGH-VALUE TO RECALL-FLG.
*--< ステージNo. 設定 >--
MOVE STAGE-NO OF CTRL-AREA TO PS.
IF (PS < 1) OR (PS > 5)
    MOVE "STAGEERR" TO MSG-NO
END-IF.
*--< 画面項目 "空" の設定 >--(LOW-VALUE / ALL"9" )
MOVE LOW-VALUE TO WK-@%1@.
*--< W03 "空" の設定 >--(LOW-VALUE / ALL"9" )
MOVE LOW-VALUE TO WK-ITEMS.
MOVE WK-@%1@ TO W03-@%1@.
*--< 初期化 >--
INITIALIZE W03-NON.
INITIALIZE W03-CNT.
INIT-EXIT.
EXIT.

* *****
*          *      位相要素      *
* *****

Y-PROC SECTION.
Y-START.
*--< @%1@
COPY @%1@.
Y-EXIT.
EXIT.

* *****
*          *      論理要素      *
* *****

```

L-PROC SECTION.

L-START.

*--< @%1@

COPY @%1@.

L-EXIT.

EXIT.

```

*
*          *****
*          *      後処理      *
*          *****

```

FIN-PROC SECTION.

FIN-START.

*--< ファイル CLOSE >

FIN-EXIT.

EXIT.

```

*
*          *****
*          *  位相要素  -  section 群  *
*          *****

```

*--< @%1@

COPY @%1@.

```

*
*          *****
*          *  論理要素  -  section 群  *
*          *****

```

*--< @%1@

COPY @%1@.

```

*
*          *****
*          *  FILE I/O      *
*          *****

```

```

*          *****

```

```

*          *  読込サブルーチン

```


*

*--< シーケンシャル READ ? @%8@ >--

IO-W03-@%7@-SEC SECTION.

IO-W03-@%7@-START.

IF @%7@-OPEN NOT = "1"

OPEN INPUT @%16@

MOVE "1" TO @%7@-OPNS

END-IF.

READ @%16@ AT END

MOVE "E" TO F-END

GO TO IO-W03-@%7@-EXIT

END-READ.

IO-W03-@%7@-EXIT.

EXIT.

*--< KEY READ ? @%8@>--

IO-W03-@%7@-SEC SECTION.

IO-W03-@%7@-START.

IF @%7@-OPNS NOT = "1"

OPEN INPUT @%16@

MOVE "1" TO @%7@-OPNS

END-IF.

IF @%7@-REDS NOT = "1"

READ @%16@ INVALID

MOVE "IV" TO @%7@-INVS

GO TO IO-W03-@%7@-EXIT

END-READ

MOVE "1" TO @%7@-REDS

IO-W03-@%7@-EXIT.

EXIT.

```

*
*****
*
*       書込サブルーチン
*
*****

```

*--< 書込み @%8@ >--

IO-W03-@%7@-SEC SECTION.

IO-W03-@%7@-START.

```

IF      @%7@-OPNS   NOT =      "1"
      OPEN   OUTPUT      @%16@
      MOVE   "1"      TO  @%7@-OPNS

```

END-IF.

```

WRITE  @%16@-REC      INVALID
      MOVE  "IV"      TO  @%7@-INVS
      GO TO      IO-W03-@%7@-EXIT

```

END-WRITE.

IO-W03-@%7@-EXIT.

EXIT.

```

*
*****
*
*       更新サブルーチン
*
*****

```

IO-W03-@%7@-SEC SECTION.

IO-W03-@%7@-START.

*--< KEY READ ? @%8@ >--

```

IF      @%7@-OPNS   NOT =      "1"
      OPEN   I-0      @%16@
      MOVE   "1"      TO  @%7@-OPNS

```

END-IF.

* --< k e y 設定

```

      MOVE   WFL-RD?-KEY-#1      TO  k e y 設定 117
      READ  @%16@      INVALID

```

```

                                MOVE "IV"    TO    @%7@-INVS
                                GO TO        IO-W03-@%7@-EXIT

                                END-READ.

*--< UPDATE ?  @%8@ >--
                                IF    @%7@-INVS    NOT =    "IV"
                                REWRITE @%16@-REC    INVALID
                                MOVE "IV"    TO    @%7@-INVS
                                GO TO        IO-W03-@%7@-EXIT

                                END-REWRITE

                                END-IF.

                                IO-W03-@%7@-EXIT.

                                EXIT.

```

*

なお、以上のプログラムにおいて、
 %24は %16+STS、
 %17は定義表のファイルの名称、
 %9は定義表のファイルの単語識別子、
 %7は定義表のファイルのR-NO、
 %1はW0n
 がそれぞれ対応している。

【0122】

以上の未定義のW03論理要素群やW03位相要素群、未定義のW03パレット関数についても同様に定義されプログラムが生成される。

【0123】

また、W02論理要素、W02パレット関数、W04論理要素、W04位相要素及びW04パレット関数についても同様に自動的にプログラムが生成される。

そして、これらの位相要素及び論理要素並びにパレット関数に係るプログラムは組込部805においてパレット連鎖関数に組み込まれ、出力部806より出力

される。

【0124】

【発明の効果】

本発明によれば、パレット関数、位相要素及び論理要素のいずれもデータエリアのアドレスだけでロジックを実現していることから、業務用ソフトウェアやゲーム用ソフトウェア、その他あらゆる分野のソフトウェア生成の自動化が可能であり、しかも生成されたソフトウェアは、生産性の向上、品質の均一化、保守の容易化が図れる。

【図面の簡単な説明】

- 【図1】 L Y E E の構成図である。
- 【図2】 パレット連鎖関数のフローチャートである。
- 【図3】 パレット関数のフローチャートである。
- 【図4】 位相要素のフローチャートである。
- 【図5】 W 0 3 論理要素のフローチャートである。
- 【図6】 W 0 2 論理要素のフローチャートである。
- 【図7】 W 0 4 論理要素のフローチャートである。
- 【図8】 自動生成装置の構成図である。
- 【図9】 ソフトウェアの生成手順を示す図である。
- 【図10】 定義体の一例を示す図である。
- 【図11】 処理経路図の一例を示す図である。
- 【図12】 定義表の一例を示す図である。
- 【図13】 変換部の動作を示すフローチャートである。

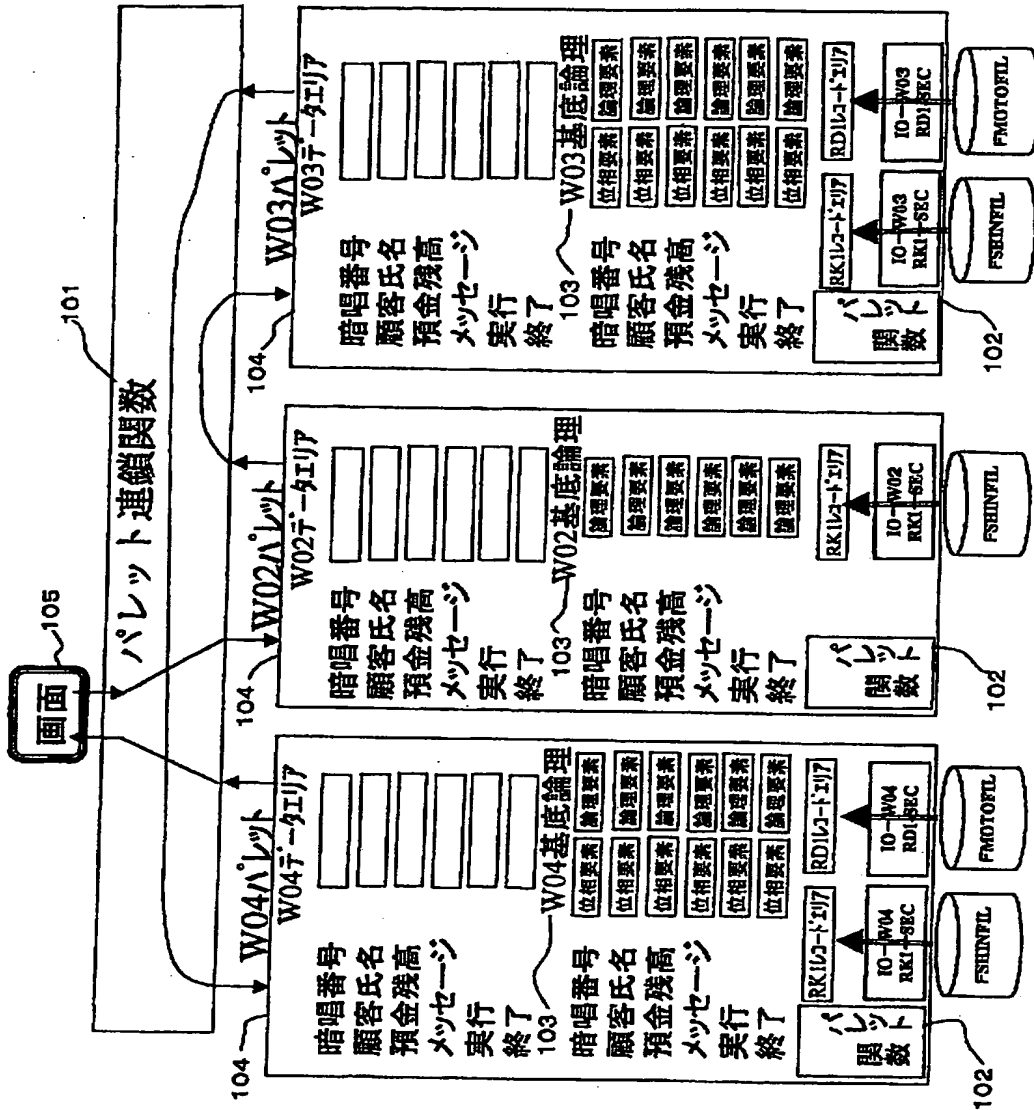
【符号の説明】

- 801 第1の記憶部
- 802 第2の記憶部
- 803 入力部
- 804 埋込部
- 805 組込部
- 806 出力部

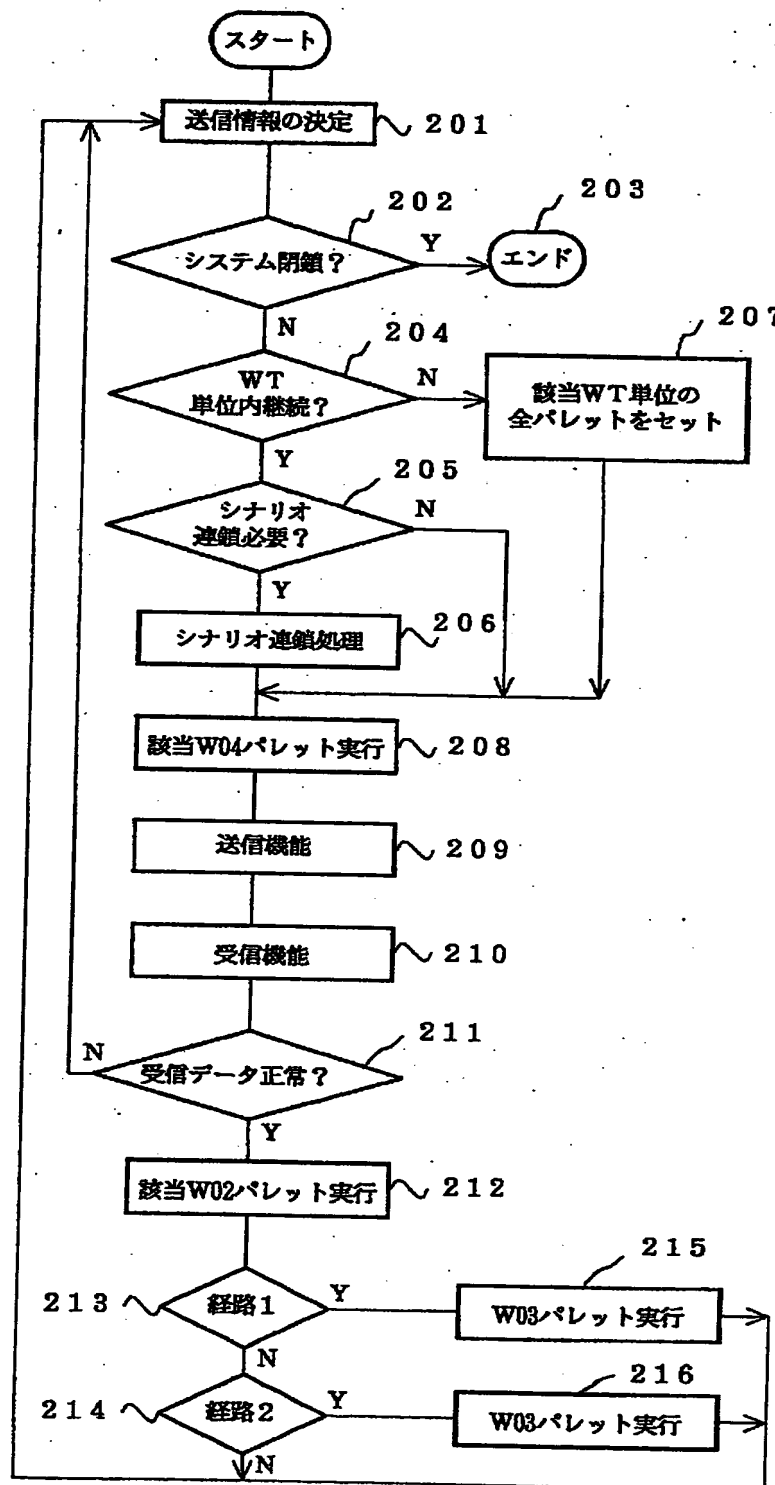
【書類名】

図面

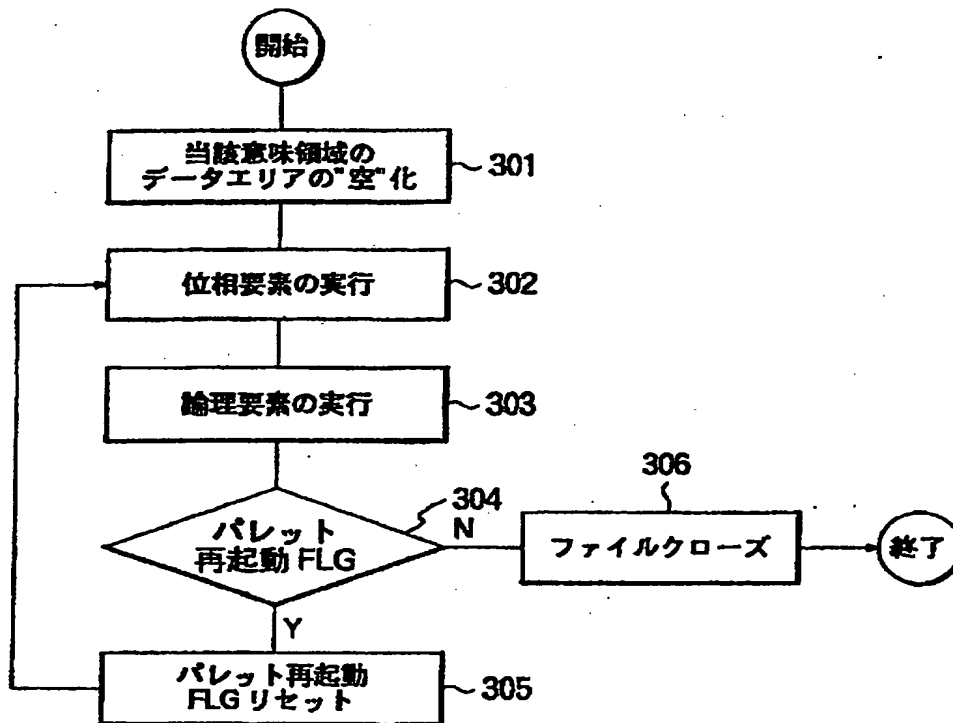
【図 1】



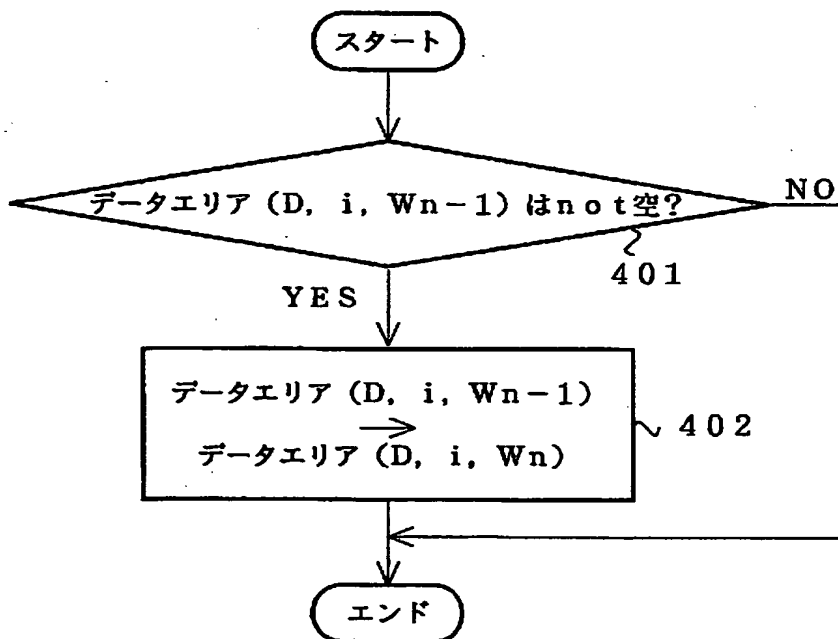
【図2】



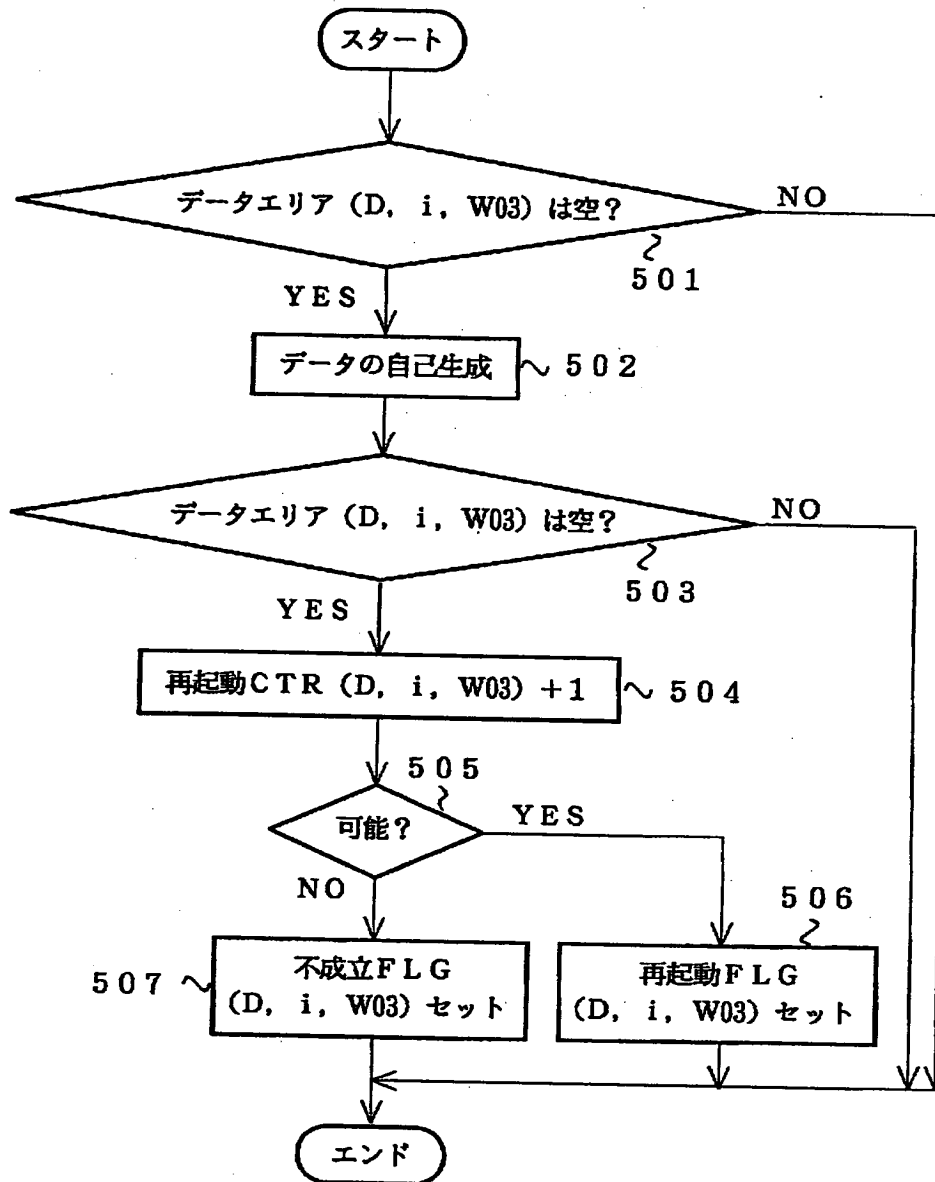
【図 3】



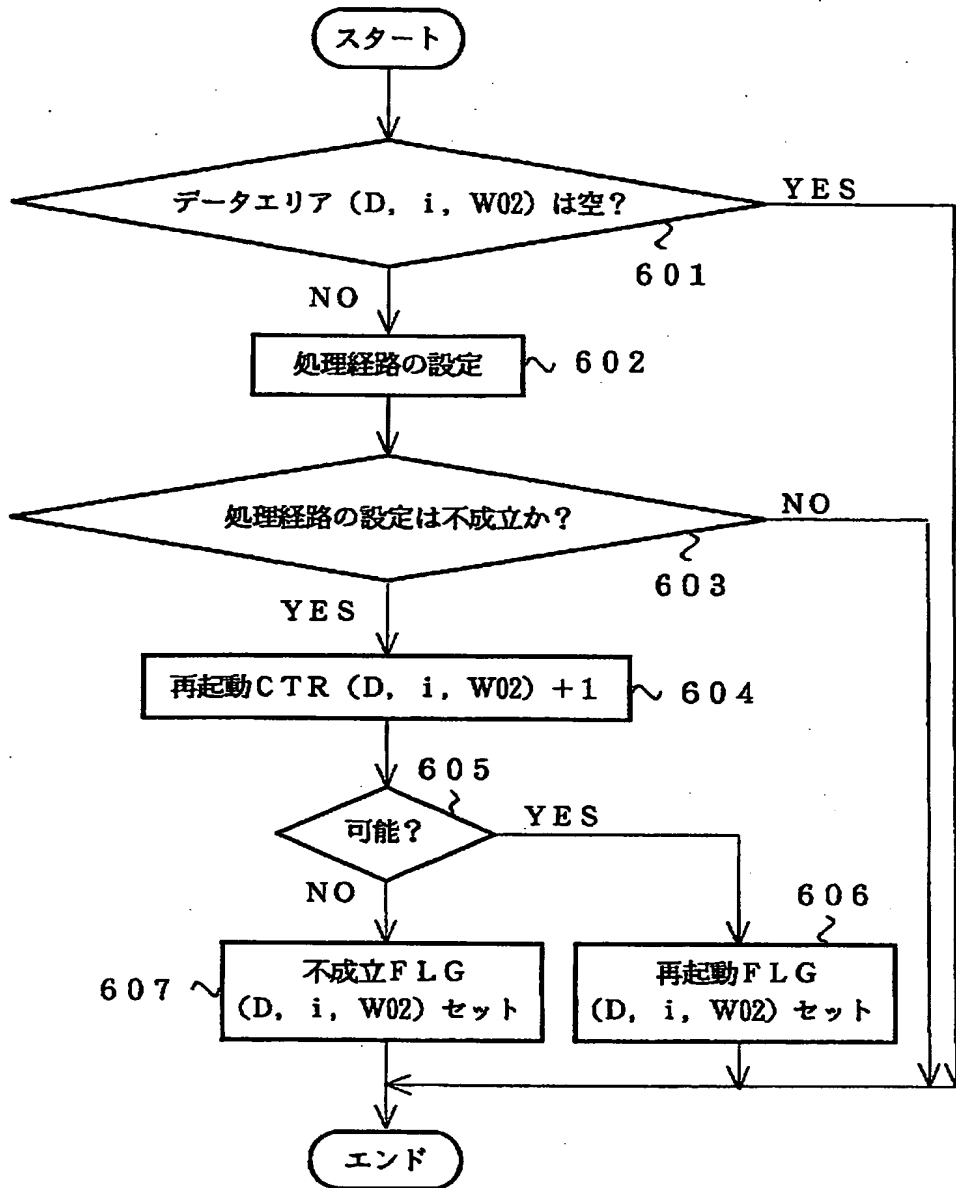
【図 4】



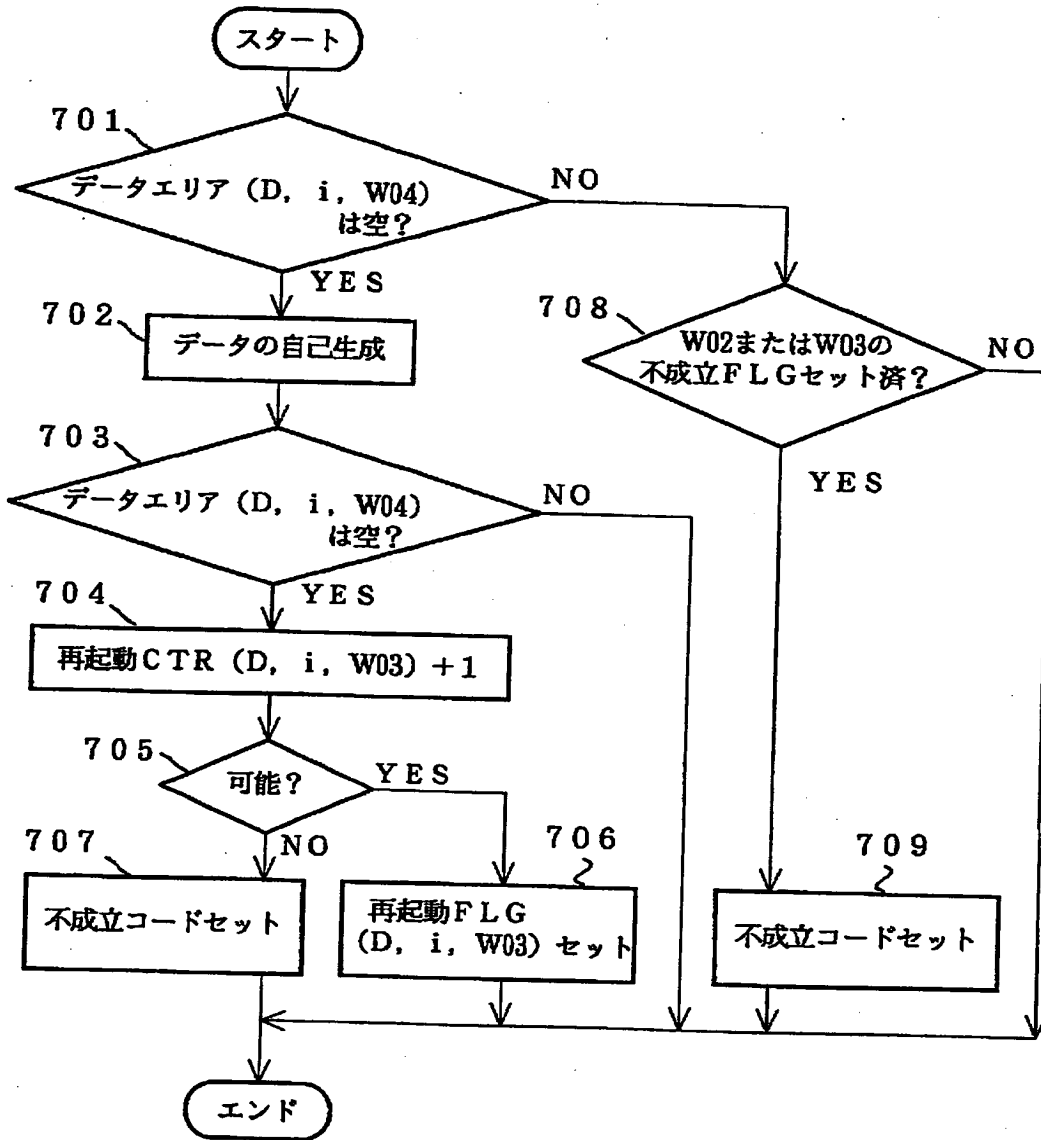
【図5】



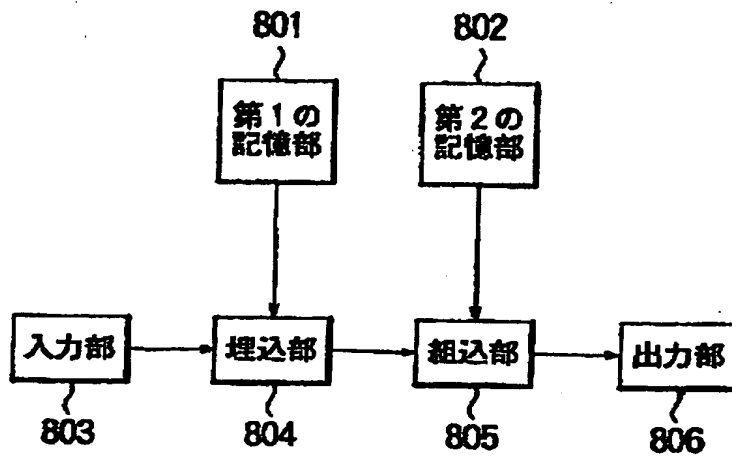
【図6】



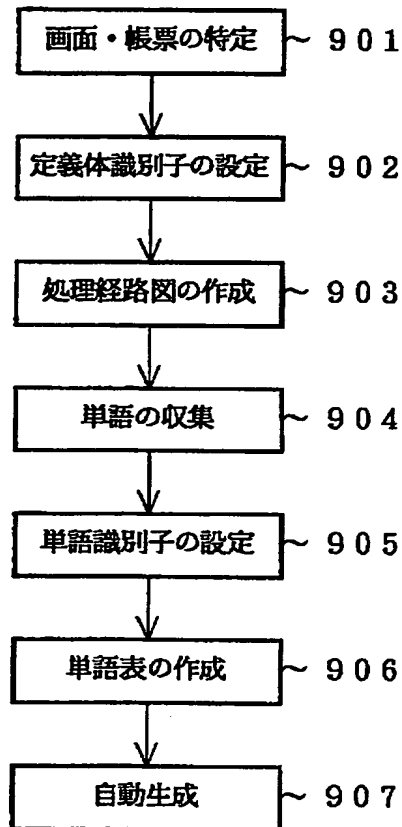
【図7】



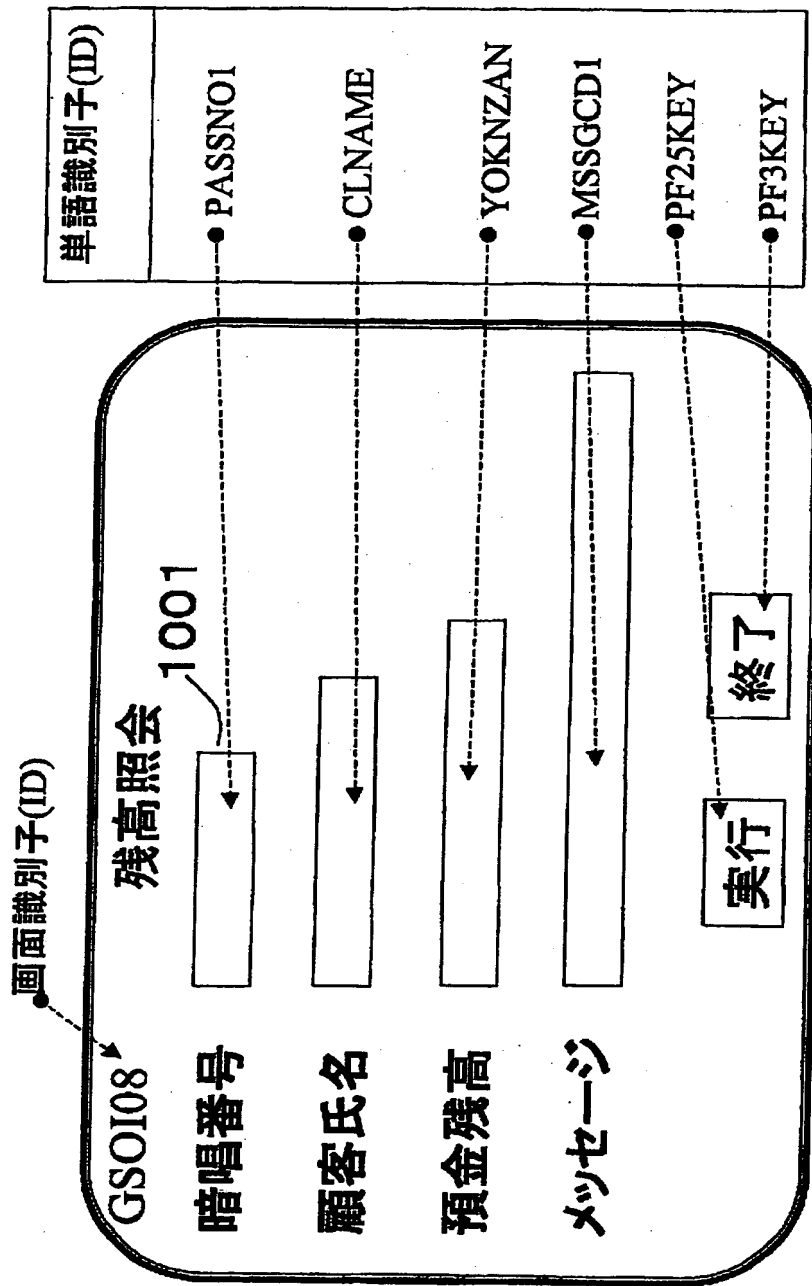
【図 8】



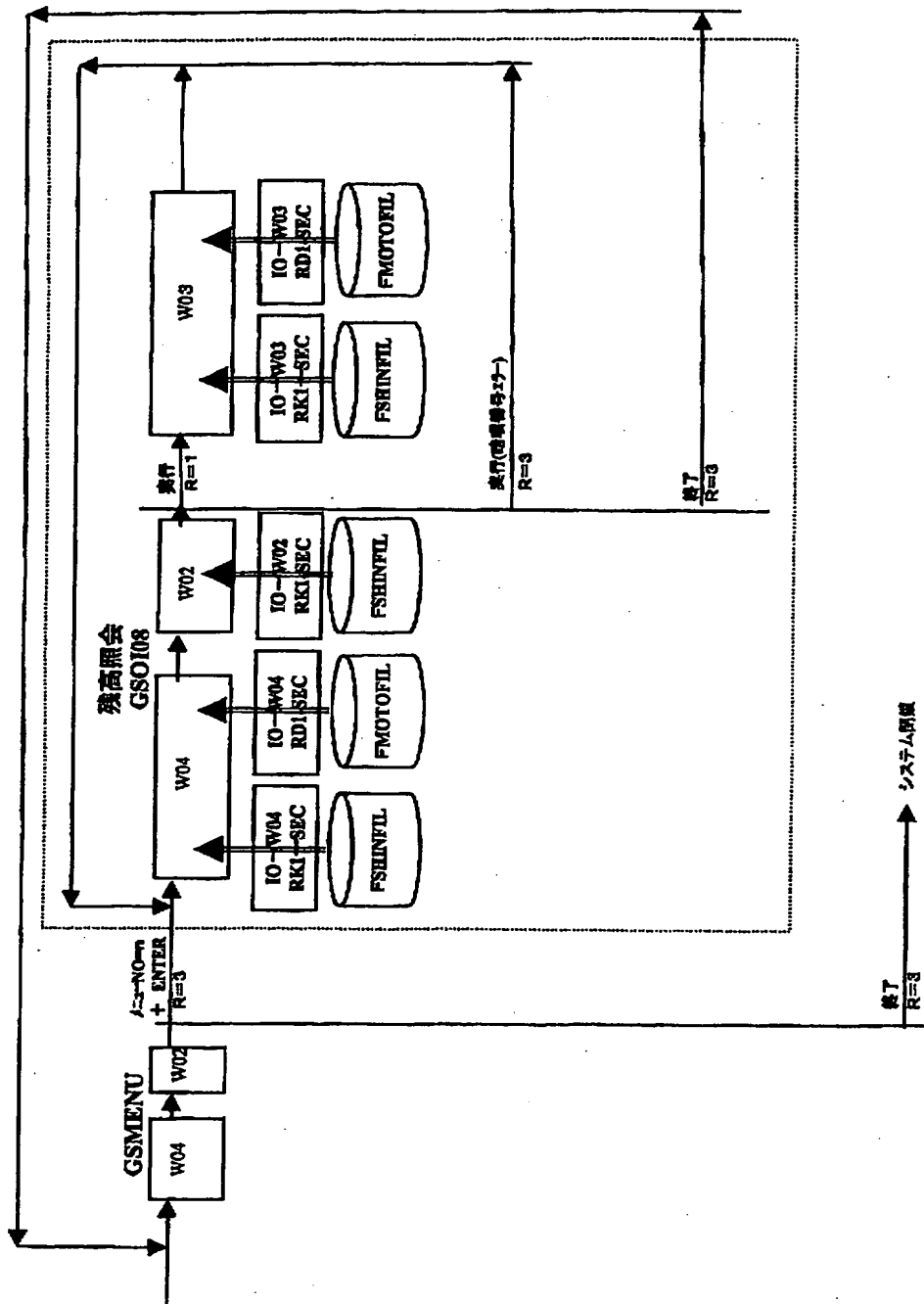
【図 9】



【図10】



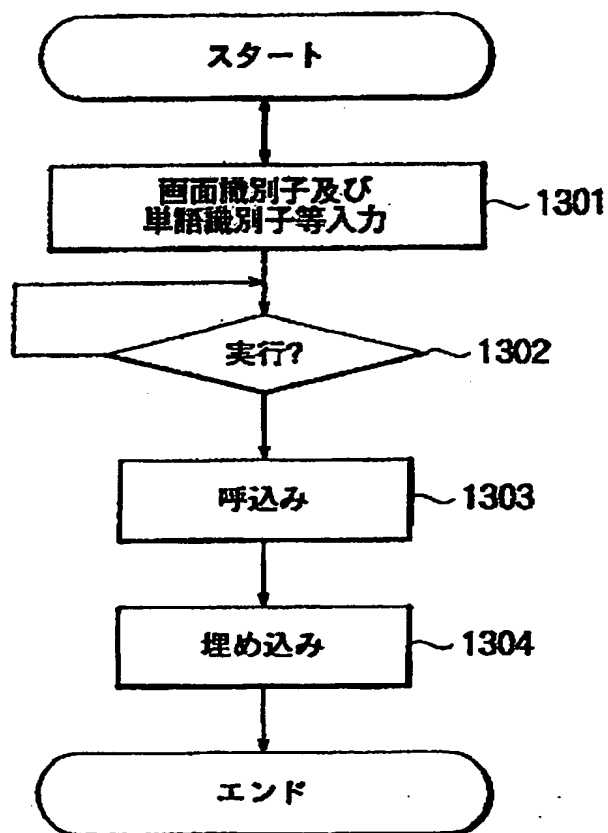
【図 1 1】



【図12】

画面ID	GSOI08	画面名称	残高照会	桁数	属性	使用 ファイル	R-NO	PF-KEY	次画面識別子	単語定義
項番	単語名称	単語識別子	I/O							
1	暗証番号	PASSN01	I	4	9	8				
2	顧客氏名	CLNAME 0	O	25	X	8				
3	預金残高	YOKNZAN	O	9	9	7				
4	メッセージ	MSSGCD1	O	50	X					
5	終了	PF3KEY		1	9		3	PF 3 KEY	GSMENU	
6	実行	PF25KEY		1	9		2	PF 2 5 KY	GSOI 0 8	
7	預金元帳(FN070FIL)	FN070-REC					ED 1			
8	顧客台帳(FSHINFIL)	FSHIN-REC					EK 1			
9										
10										

【図 13】



【書類名】 要約書

【要約】

【課題】 ソフトウェアをSEの経験、知識、応用力等の属人的能力を必要とせず機械的アルゴリズムだけで自動的に生成すること。

【解決手段】 L Y E E理論によるソフトウェアを生成するにあたり、定義体識別子、単語識別子、処理経路及び設定値を導出するための導出データが未定義のW04位相要素、W04論理要素、W02論理要素、W03位相要素及びW03論理要素、パレット関数を予め記憶しておき、単語毎に定義体識別子、単語識別子、処理経路及び導出データを入力し、単語毎に入力された定義体識別子、単語識別子、処理経路及び導出データを、未定義のW04位相要素、W04論理要素、W02論理要素、W03位相要素、W03論理要素及びパレット関数の所定の位置に埋め込む。

【選択図】 図8

【書類名】 職権訂正データ
【訂正書類】 特許願

<認定情報・付加情報>

【特許出願人】
【識別番号】 598037422
【住所又は居所】 神奈川県鎌倉市十二所 9 6 7 - 6 4
【氏名又は名称】 根来 文生
【代理人】 申請人
【識別番号】 100077849
【住所又は居所】 東京都千代田区神田多町 2 丁目 1 番地 神田東山ビル
【氏名又は名称】 須山 佐一

出 願 人 履 歴 情 報

識別番号

[598037422]

1. 変更年月日 1998年 3月20日

[変更理由] 新規登録

住 所 神奈川県鎌倉市十二所967-64

氏 名 根来 文生

特平10-071838

出 願 人 履 歴 情 報

識別番号 [598153401]

1. 変更年月日 1998年10月 2日
[変更理由] 新規登録
住 所 東京都港区高輪三丁目11番3号
氏 名 株式会社アイエスデー研究所

